



## Creando listas con Objetos

### Transcripción

[00:00] Hola. ¿Cómo están? En las clases anteriores hemos visto el bloque de trabajando con ArrayList. Ahora, en la clase 4 vamos a ver listas de objetos. La primera clase va a ser relacionada a creando listas con objetos. Para eso vamos a hacer de nuevo duplicar la clase 3 y colocar clase 4.

[00:25] Recordando también que aquí el package hizo rename para com.alura. Pueden poner cualquier otro nombre pero básicamente estoy utilizando este padrón ahora. Una vez hecho esto, vamos a crear una clase nueva. Nuestra clase la vamos a llamar curso.

[00:51] En nuestra clase curso vamos a colocar nuestras variables nombre y un entero de tiempo. Esas clases vamos a generar nuestro getters and setters y también vamos a generar nuestro constructor, que va a servir para crear un objeto del tipo curso, un constructor ya pasando los parámetros en nombre y tiempo.

[01:46] Una vez hecho esto, vamos a remover todo lo que estaba aquí y vamos a hacer lo siguiente. Vamos a colocar curso1. En este curso vamos a importar la clase curso, vamos a crear un nuevo objeto del tipo curso y vamos a colocar nuestras variables, que son nombre del curso y el tiempo de duración, vamos a ponerle tiempo en horas por ejemplo, entonces este curso va a demorar 30 horas.

[02:31] Vamos a hacer lo mismo con otros cursos, vamos a crear tres cursos más, colocar curso2, curso3, curso 4. Aquí vamos a colocar geometría, vamos

colocar aquí física, vamos a colocar aquí química, vamos a colocar aquí física.  
No, física ya colocamos, vamos a colocar aquí historia.

[03:13] En tiempos vamos a colocarle aquí 10, aquí 20 y aquí 50 horas por ejemplo. Una vez hecho esto, imaginémonos que aquí quiero adicionar ArrayList de tipo string, quiero adicionar los cursos. Si yo hago aquí lo siguiente, aparecerá en rojo. ¿Por qué? Porque aquí estoy diciendo que el ArrayList es del tipo string.

[03:45] Para eso yo tendría que cambiar el ArrayList y tendría que ponerlo que sea del tipo curso. Ahí yo voy a poder adicionar cursos, después pongo curso2, curso3, curso4. Una vez hecho esto, vamos a imprimir nuestros cursos, System.out.println y vamos a colocar por ejemplo cursos.

[04:25] ¿Qué va a imprimir aquí? ¿Aquí qué está imprimiendo? Aquí está imprimiendo en realidad el objeto. ¿Por qué? Porque él está utilizando la implementación de la clase toString. Internamente él implementa el toString, solamente que el toString del objeto es prácticamente el com.alura.Curso y este de aquí sería el dónde está ubicado el objeto en memoria.

[04:54] Pero yo no quiero esto, yo quiero solamente por ejemplo quiero colocar el nombre del curso. Entonces para esto vamos a hacer aquí implementar el override y aquí implementar el public, el método toString. ¿Ese método toString qué me devuelve? Un string. Entonces aquí vamos a colocar return y vamos a colocar por ejemplo this.nombre.

[05:39] Vamos a imprimir nuevamente. ¿Qué imprimió? Geometría, física, química, historia. Entonces aquí estoy sobrescribiendo el método toString que ya viene por defecto en todas las clases.

[06:00] Solamente que si no lo utilizo, él va a imprimir el toString del objeto que yo estoy creando, por eso aparecen esos valores extraños que ustedes están viendo o que nosotros vimos. Esto básicamente sería en nuestra clase 4. Hasta la próxima.

